

一种求倒数近似值的量子算法及其量子电路

朱家良,叶 宾,季 雯

(中国矿业大学 信息与控制工程学院,江苏 徐州 221116)

摘 要:求取一个无符号数的倒数在数值计算中有着重要的应用。如何在量子电路中高效准确地求出倒数,影响着许多量子算法的性能。在此提出了一种求倒数近似值的量子算法及其量子电路的设计方法。首先将输入的二进制数存储在输入寄存器中;通过添加 Toffoli 门将两个 n 位二进制数每一位相乘的结果保存在 $2n$ 个辅助量子比特中;再重复利用基础量子门设计出的 n 量子全加器对辅助量子比特进行低位置零的移位相加;用控制非门设计置零电路对辅助寄存器进行置零操作以重复利用辅助量子比特,最后设计出了一种量子电路宽度较小的量子乘法器。应用牛顿迭代法解得一个求倒数近似值的系统图,然后在上述量子全加器和量子乘法器的基础上,设计出系统中各模块的量子电路图,最后连接各模块电路图,形成一个完整的求倒数量子算法的量子电路。通过分析,该量子电路提高了辅助量子比特的利用率,并且具有较低的计算复杂性。

关键词:量子算法;量子电路;倒数;量子全加器;量子乘法器

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2022)03-0090-06

doi:10.3969/j.issn.1673-629X.2022.03.015

A Quantum Algorithm for Finding Reciprocal Approximation and Its Quantum Circuit

ZHU Jia-liang, YE Bin, JI Wen

(School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China)

Abstract: Finding the reciprocal of an unsigned number plays an important role in numerical calculations. How to find the reciprocal efficiently and accurately in quantum circuits affects the performance of many quantum algorithms. A quantum algorithm is proposed to calculate the approximation of the reciprocal and the corresponding quantum circuits are designed. Firstly, the input binary number is stored in the input register. Then, the multiplication results of two n -bit binary numbers are stored in $2n$ auxiliary qubits by adding Toffoli gate. After the least significant auxiliary bits are set to 0, the n -bit quantum full adder designed by the basic quantum gate is reused to shift and add the auxiliary qubits. The zero setting circuit is designed by the CNOT gate to zero the auxiliary register, so the auxiliary qubit can be reused. Finally, a multiplier with small quantum circuit width is designed. Newton iterative method is used to obtain a system diagram for reciprocal approximation. Then based on the above-mentioned quantum full adder and quantum multiplier, each module of the system diagram is designed. Finally, each module is connected to form a complete reciprocal quantum circuit. Through analysis, it can be seen that the quantum circuit improves the utilization rate of auxiliary qubits and has low computational complexity.

Key words: quantum algorithm; quantum circuit; reciprocal; quantum adder; quantum multiplier

0 引 言

量子计算以其独特的并行计算特性,在机器学习、高维数据挖掘、大数据处理、图像处理等领域有着巨大优势^[1-4]。和传统的数字计算相比较,量子计算显著减少了执行时间和能耗^[5]。量子计算机的基本存储单元是量子比特,一个量子比特可表示量子态 $|0\rangle$ 和 $|1\rangle$ 的叠加,因此一次运算就可同时处理两个基态。处理 2^n 状态的信息, n 个量子比特仅需一次操作,远快于 2^n 个经

典比特所需的 2^n 次操作^[6]。正因如此,量子计算能够指数级加速传统算法,被认为是解决算力不足的有效方法之一。

求取一个无符号数的倒数在求解量子线性系统、量子相位估计、量子支持向量机等领域有着广泛的应用。在传统算法中常用牛顿迭代法、割线法等求取倒数的近似值,但是需要多次的迭代计算。量子计算的特性可以显著降低牛顿迭代计算的问题规模。Cao^[7]于2007年

收稿日期:2021-03-27

修回日期:2021-07-28

基金项目:江苏省自然科学基金项目(BK20191339);河南省网络密码技术重点实验室研究课题(LNCT2019-S06)

作者简介:朱家良(1999-),男,研究方向为量子计算;通信作者:叶 宾(1980-),男,副教授,硕导,研究方向为量子计算。

首次提出了一种基于牛顿迭代法近似求取正整数倒数的量子线路,但其耗费辅助量子比特较多,线路结构复杂。该文提出一种更简洁的近似求取无符号数倒数的量子算法,此算法以基本的量子全加器^[8-9]和量子乘法器^[10-11]为基础模块,通过少量辅助寄存器进行连接,经过移位操作等搭建出量子电路。经过分析,其时间复杂度为 $O(n^3)$,空间复杂度为 $O(2n^3)$,具有量子代价小、结构简单等特点,有利于降低构造量子电路的成本及物理实现难度。

1 量子逻辑门基础

量子计算中的基本单位量子比特(qubit)与经典计算机中的比特(bit)相对应,单量子比特的状态为2个基态相叠加的形式^[12]: $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$,其中 α 和 β 均为复数,满足 $|\alpha|^2 + |\beta|^2 = 1$ 。

与经典计算机中的逻辑门类似,量子计算中的计算单元由量子逻辑门组成,它们在数学上是一个么正变换和酉变换,并且这个变换是可逆的,在此介绍几种常用的量子逻辑门:

非(NOT)门,是一个单量子门,量子电路符号如图1(a)所示,它的作用是使量子态从 $|0\rangle$ 向 $|1\rangle$ 的翻转。

Hadamard门又称H门,它的作用是将单量子态转变为叠加态,如图1(b)所示,它实现将 $|0\rangle$ 状态转变为 $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$, $|1\rangle$ 状态转变为 $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ 。

受控非(CNOT)门^[13],是一个双量子逻辑门,由控制位和目标位组成。如图1(c)所示,它的作用是若控制位的量子态为 $|0\rangle$,则目标位上的量子态不发生翻转;反之,若控制位量子态为 $|1\rangle$,目标位上的量子态发生翻转。

Toffoli门^[14],是一个三量子比特门,由两个控制位和一个目标位组成。如图1(d)所示,仅当控制位 $|a\rangle$ 和 $|b\rangle$ 都为 $|1\rangle$ 时,目标位 $|c\rangle$ 中的状态发生翻转,否则不发生变化。

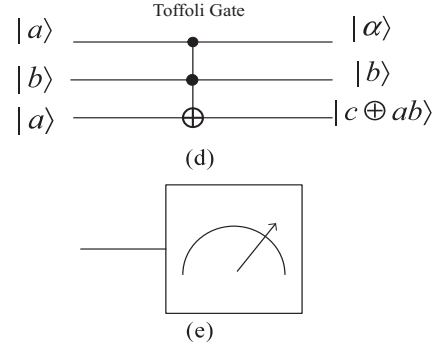
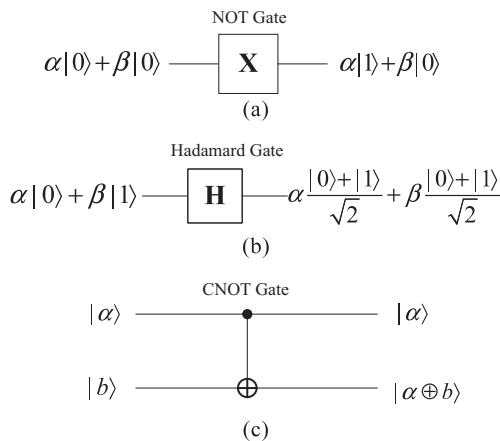


图1 量子逻辑门

测量门,将量子比特投影到经典比特中,实现观测。如图1(e)所示输入为处于叠加态的量子位,输出投影到 $|0\rangle$ 和 $|1\rangle$ 下的概率幅度。

2 量子乘法器的实现

2.1 量子半加器

n 位的量子全加器是由 n 个半加器组成的,该文设计的半加器包括1个Toffoli门和2个CNOT门,如图2所示。其中 $|a\rangle$ 和 $|b\rangle$ 是输入的二进制数, $|0\rangle$ 是辅助量子位,结果存储在 $|\text{sum}\rangle$ 中, $|c_{\text{out}}\rangle$ 是进位。

半加器的过程如下:

$$|a\rangle|b\rangle|0\rangle|0\rangle \rightarrow |a\rangle|b\rangle|\text{sum}\rangle|c_{\text{out}}\rangle \quad (1)$$

辅助量子位输出:

$$|0\rangle|0\rangle \rightarrow |\text{sum}\rangle|c_{\text{out}}\rangle \quad (2)$$

$$|\text{sum}\rangle = |a \oplus (b \oplus 0)\rangle \quad (3)$$

$$|c_{\text{out}}\rangle = |ab \oplus 0\rangle \quad (4)$$

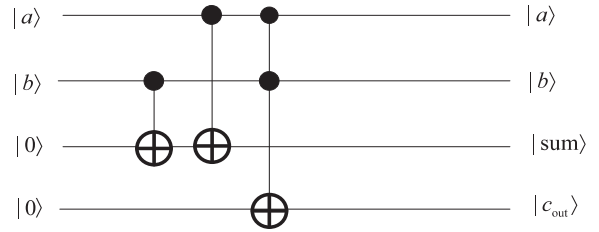


图2 量子半加器

2.2 n 位量子全加器

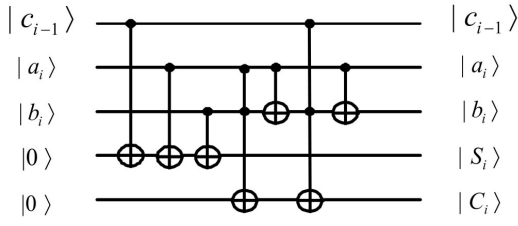
量子全加器是量子乘法器实现的基础。2002年,Cheng Kai-Wen^[8]首次完善了量子全加器和量子全减器的电路图。常丽等人^[9]于2019年采用了超前进位方式,简化了量子全加器的结构。在此基础上,该文根据全加器的逻辑表达式设计出单比特的量子全加器电路图,如图3(a)所示(图3(b)为其简化图)。然后将多个单比特的量子全加器组合,得到 n 比特量子全加器,如图3(c)所示。

$$|s_i\rangle = |a_i \oplus b_i \oplus c_{i-1}\rangle \quad (5)$$

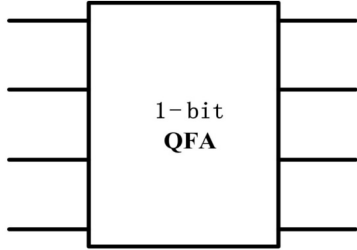
$$|c_i\rangle = |a_i b_i \oplus (a_i \oplus b_i) c_{i-1}\rangle \quad (6)$$

其中, $|a_i\rangle$ 是一个二进制数 a 的第 i 位, $|b_i\rangle$ 是另一个二进制数 b 的第 i 位, $|c_i\rangle$ 是进位 c 的第 i 位, $|s_i\rangle$ 是 a

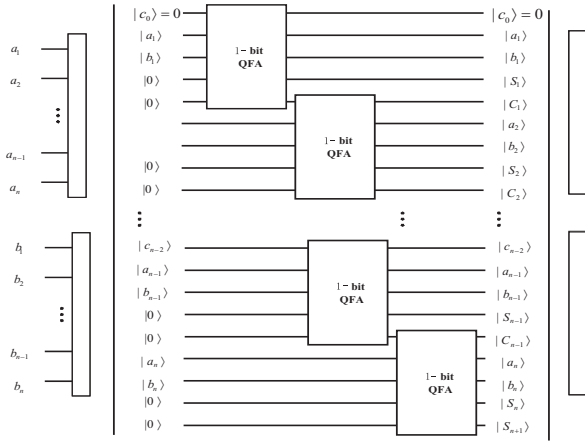
与 b 之和的第 i 位。



(a) 单比特量子全加器电路图



(b) 图(a)的简化图



(c) n 比特量子全加器电路图

图3 量子全加器

2.3 量子乘法器

目前大多数的量子乘法器是在一个 n 量子比特的寄存器中输入 $|a\rangle_1 = |a_{n-1}\rangle_1 \otimes |a_{n-2}\rangle_1 \otimes \cdots \otimes |a_0\rangle_1$ 来表示整数 $a = 2^{n-1}a_{n-1} + 2^{n-2}a_{n-2} + \cdots + 2^0a_0$; 在另一个 n 量子比特的寄存器中输入 $|b\rangle_2 = |b_{m-1}\rangle_2 \otimes |b_{m-2}\rangle_2 \otimes \cdots \otimes |b_0\rangle_2$ 来表示整数 $b = 2^{m-1}b_{m-1} + 2^{m-2}b_{m-2} + \cdots + 2^0b_0$; 第三个 l 量子比特的寄存器用来存放前两个寄存器中的数相乘的结果, 其状态为 $|c\rangle_3 = |c_{l-1}\rangle_3 \otimes |c_{l-2}\rangle_3 \otimes \cdots \otimes |c_0\rangle_3$, 其中 $c = 2^{l-1}c_{l-1} + 2^{l-2}c_{l-2} + \cdots + 2^0c_0$ 。经过量子乘法器运算后各寄存器中的数值改变如下:

$$|a\rangle_1 |b\rangle_2 |c\rangle_3 \rightarrow |a\rangle_1 |b\rangle_2 |c + ab \bmod 2^l\rangle_3 \quad (7)$$

最后通过测量第三个量子寄存器, 将其读数写入经典寄存器来得到两数相乘的结果^[15-17]。

2.4 量子乘法器的移位实现

在实现量子乘法运算的过程中, 最重要的一步就是对乘数进行移位相加, 例如实现 110 和 111 的乘积, 其实

现步骤见图 4(a)。由其规律可知, 部分积之间的相加要实现左移操作, 即在量子计算机上实现移位寄存器。文献[10]中设计的量子乘法器完善了移位寄存器的功能, 算法整体的时间复杂度为 $O(n^2)$, 空间复杂度为 $O(n^2)$ 。Suzuki^[11]在 2020 年提出的量子乘法器的时间复杂度和空间复杂度都较低, 但是结构复杂。该文运用对寄存器低位置零的操作达到移位寄存器的效果, 拥有结构简单、复杂度低等优点。具体步骤如下:

用寄存器 $|0\rangle_s = |0\rangle_s^{(m-1)} \otimes |0\rangle_s^{(m-2)} \otimes \cdots \otimes |0\rangle_s^{(0)}$ 来存放部分积的数值 (s 代表第 s 个部分积, m 代表两个乘数中位数较多的二进制数的位数), 例如上述例子中 110 与 111 的三个部分积分别表示为:

$$\begin{aligned} |0\rangle_1 &= |0\rangle_1^{(m-1)} \otimes |0\rangle_1^{(m-2)} \cdots \otimes |0\rangle_1^{(0)} \rightarrow |1\rangle_1 = |1\rangle_1^2 \otimes |1\rangle_1^1 \otimes |0\rangle_1^0 \\ |0\rangle_2 &= |0\rangle_2^{(m-1)} \otimes |0\rangle_2^{(m-2)} \cdots \otimes |0\rangle_2^{(0)} \rightarrow |0\rangle_2 = |1\rangle_2^2 \otimes |1\rangle_2^1 \otimes |0\rangle_2^0 \\ |0\rangle_3 &= |0\rangle_3^{(m-1)} \otimes |0\rangle_3^{(m-2)} \cdots \otimes |0\rangle_3^{(0)} \rightarrow |0\rangle_3 = |1\rangle_3^2 \otimes |1\rangle_3^1 \otimes |0\rangle_3^0 \end{aligned} \quad (8)$$

如图 4(b) 所示, 将图 4(a) 中部分积的低位置零, 再与高位相加, 即可实现与移位同样的效果, 110 与 111 相乘的部分积低位置零后表示为:

$$\begin{aligned} |0\rangle_1 &= |0\rangle_1^{(m-1)} \otimes |0\rangle_1^{(m-2)} \cdots \otimes |0\rangle_1^{(0)} \rightarrow |0\rangle_1 = |1\rangle_1^2 \otimes |1\rangle_1^1 \otimes |0\rangle_1^0 \\ |0\rangle_2 &= |0\rangle_2^{(m-1)} \otimes |0\rangle_2^{(m-2)} \cdots \otimes |0\rangle_2^{(0)} \rightarrow |1\rangle_2 = |1\rangle_2^3 \otimes |1\rangle_2^2 \otimes |0\rangle_2^1 \otimes |0\rangle_2^0 \\ |0\rangle_3 &= |0\rangle_3^{(m-1)} \otimes |0\rangle_3^{(m-2)} \cdots \otimes |0\rangle_3^{(0)} \rightarrow |0\rangle_3 = |1\rangle_3^4 \otimes |1\rangle_3^3 \otimes |0\rangle_3^2 \otimes |0\rangle_3^1 \otimes |0\rangle_3^0 \end{aligned} \quad (9)$$

分析可知, 此方法耗费寄存器的空间复杂度仅为 $O(2n-1)$ (n 为两个乘数中位数较大的二进制数的位数)。

$$\begin{array}{r} 110 \\ \times 111 \\ \hline 110 \\ 110 \\ 110 \\ \hline 101010 \end{array}$$

(a) 移位乘法步骤

$$\begin{array}{r} 110 \\ \times 111 \\ \hline 110 \\ 1100 \\ 11000 \\ \hline 101010 \end{array}$$

(b) 低位置零乘法步骤

图4 置零乘法的实现步骤

2.5 量子乘法器的实现

下面按照图 3(b) 中所示的乘法计算过程,将输入寄存器、置零寄存器、全加器,以及输出寄存器(包含量子寄存器和经典寄存器)合并以设计出完整的量子乘法器。

如图 5 所示,输入寄存器中的 n 位二进制数 $|a\rangle_1 = |a\rangle_1 \otimes |a\rangle_2 \otimes \cdots \otimes |a\rangle_n$ 和 m 位二进制数 $|b\rangle_2 = |b\rangle_1 \otimes |b\rangle_2 \otimes \cdots \otimes |b\rangle_m$ 相乘,将部分积 $|a_0 b_t\rangle, |a_1 b_t\rangle \cdots |a_n b_t\rangle, (t = 0, 1, \cdots, m-1)$ 的结果分别存放在置零寄存器:

$$|0\rangle_0 = |0\rangle^0 \otimes \cdots \otimes |0\rangle^{t_0} \otimes \cdots \otimes |0\rangle^{m-1} (t_0 = 0),$$

$$|0\rangle_1 = |0\rangle^0 \otimes \cdots \otimes |0\rangle^{t_1} \otimes \cdots \otimes |0\rangle^{m-1} (t_1 = 1)$$

$$\vdots$$

$|0\rangle_l = |0\rangle^0 \otimes \cdots \otimes |0\rangle^{t_l} \otimes \cdots \otimes |0\rangle^{m-1} (t_l = n, l = 2 \times \max(n, m))$ 上(第 s 个置零寄存器从第 t_s 位开始存放, $s = 0, 1, \cdots, n$);再将置零寄存器与全加器相连,依次求出部分积 $|0\rangle_0, |0\rangle_1, \cdots, |0\rangle_l$ 的和(此过程中可以通过对 $|0\rangle_s$ 寄存器逐个清零来减少寄存器的使用,大大降低空间复杂度),并将结果存放在寄存器 $|c\rangle_3 = |c_0\rangle_3 \otimes \cdots \otimes |c_l\rangle_3$ 中($l = 2 \times \max(n, m)$);最后通过测量将量子寄存器上的信息转化至经典寄存器 c 上,实现结果的观测。

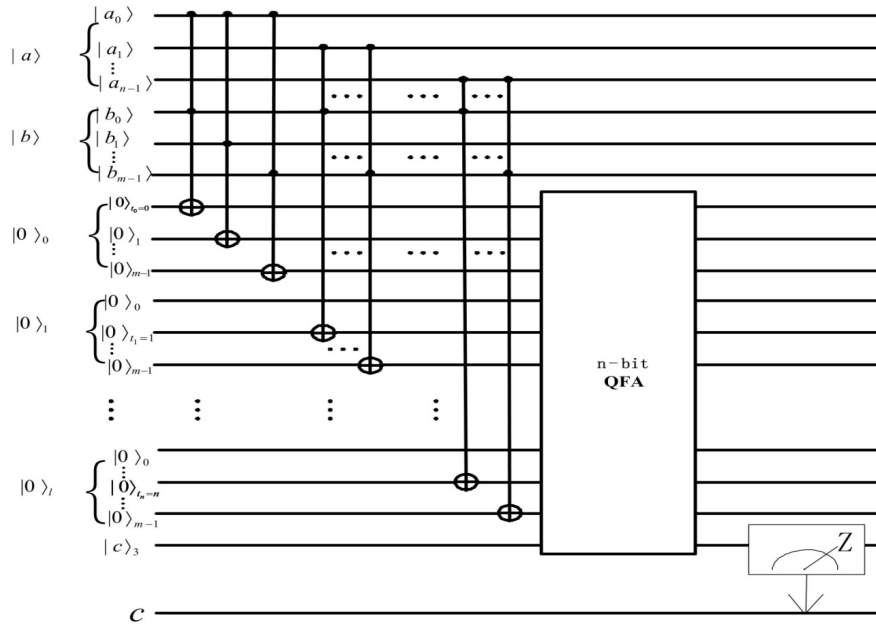


图 5 量子乘法器电路图

2.6 量子乘法器复杂度分析

为了验证算法的有效性,该文用通用量子门作为时间单元,辅助量子比特作为空间单元,分析基于牛顿迭代法的量子倒数器的时间复杂度以及空间复杂度。

首先是时间复杂度的分析。对于 n 位二进制数乘以 m 位二进制数,该文改进的量子乘法器线路如图 5 所示。在量子乘法器中,首先是用 Toffoli 门进行各位二进制数的相乘操作,需要 mn 个 Toffoli 门。随后将辅助量子寄存器中的数进行相加,由图 3(a) 可知,一个比特的量子全加器由 2 个 Toffoli 和 5 个 CNOT 门组成,因此 n 比特的量子全加器需要 $7n$ 个通用量子门。进行 m 次加法操作,因此需要用 $7mn$ 个通用量子门。因此整个乘法器需要用到 $8mn$ 个通用量子门,时间复杂度为 $O(mn)$ 。若计算的二进制数都是 n 位,则时间复杂度为 $O(n^2)$ 。

其次是空间复杂度的分析。由图 5 所知,对于 n 位二进制数乘以 m 位二进制数($m > n$),需要 $m + n$ 个输入寄存器来储存输入的二进制数。一个辅助量子

寄存器需要 m 个量子比特, m 个辅助量子寄存器一共需要 $2m^2$ 量子比特。所以量子乘法器一共需要 $2m^2 + m + n$ 个量子比特,时间复杂度为 $O(m^2)$ 。若计算的二进制数都是 n 位,则时间复杂度为 $O(n^2)$ 。

该量子乘法器需要用到 $8mn$ 个通用量子门,时间复杂度为 $O(n^2)$ ($m = n$ 时)。所需量子寄存器的数量是 $2m^2 + m + n$,空间复杂度为 $O(n^2)$ ($m = n$ 时)。

3 求倒数近似值的量子算法及其量子电路

3.1 牛顿迭代法

根据牛顿迭代公式:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (10)$$

若求取已知数 λ 的倒数,则令函数 $f(x)$ 为:

$$f(x) = \frac{1}{x} - \lambda \quad (11)$$

对 $f(x)$ 求导得:

$$f'(x) = -x^{-2} \quad (12)$$

联立式(6)~式(8)可得:

$$x_{n+1} = x_n + \frac{x_n^{-1} - \lambda}{x_n^{-2}} = -\lambda x_n^2 + 2x_n \quad (13)$$

其中, λ 为所求倒数的原数, 要求满足 $\lambda > 1$; x_0 为满足 $x_0 > 0$ 且与 λ^{-1} 相近的二进制数。图6为牛顿迭代法的电路图。

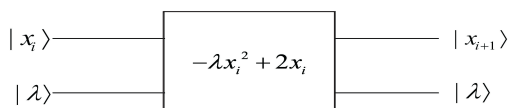


图6 利用牛顿迭代法求倒数的电路图

3.2 求倒数近似值的量子算法及其量子电路实现步骤

按照式(9)与图6的系统图, 利用3个量子乘法器和2个量子全加器可设计出单步牛顿迭代法对应的量子电路, 然后将每一步的迭代电路相连实现整个迭代过程。详细的迭代步骤如下:

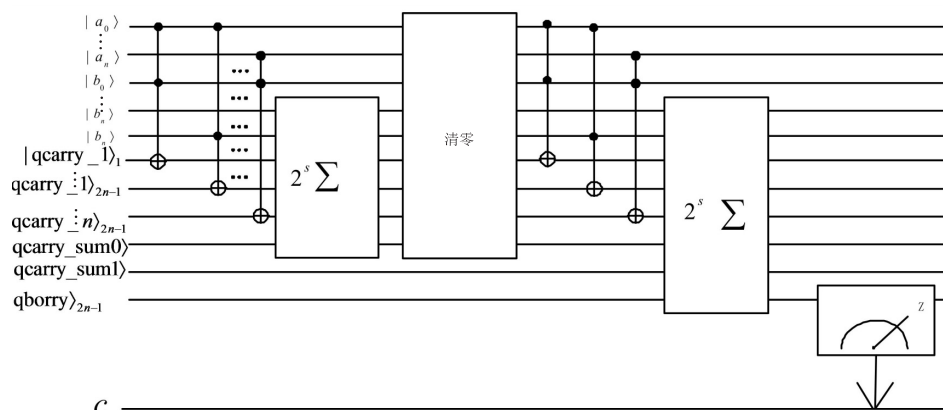
(1) 进行第一次迭代, 将2和 x_0 分别转化为二进

制数, 输入量子寄存器 $|a\rangle = |a_{n-1}\rangle \otimes |a_{n-2}\rangle \otimes \cdots \otimes |a_0\rangle$ 和 $|b\rangle = |b_{m-1}\rangle \otimes |b_{m-2}\rangle \otimes \cdots \otimes |b_0\rangle$ 中, 把 $|2x_0\rangle_0$ 的结果放入辅助寄存器 $|qcarry_sum0\rangle_{2n}$ (n 代表两个乘数中位数较多的二进制数的位数);

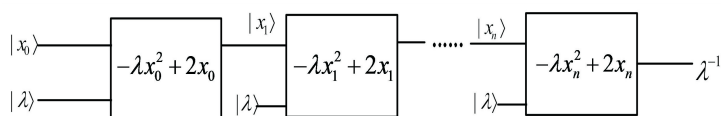
(2) 以同样的步骤求得 $\lambda \times x_0 \times x_0$ 的结果, 放入辅助寄存器 $|qcarry_sum1\rangle_{2n}$; 将 $|qcarry_sum0\rangle_{2n}$ 和 $|qcarry_sum1\rangle_{2n}$ 充当两个加数分别与全加器的两端相连, 求出 $|-\lambda x_0^2 + 2x_0\rangle_0$, 放入寄存器 $|qborrow\rangle$ 中;

(3) 通过测量将量子寄存器 $|qborrow\rangle$ 上的信息转化至经典寄存器上, 实现第一次迭代结果的观测。量子电路图如图7(a)所示;

(4) 将上一步得到的结果作为新的输入 x_1 , 以同样的方法求出 $|-\lambda x_1^2 + 2x_1\rangle_1$ 。将 n 个这样的电路迭代相连 (n 表示迭代的次数), 如图7(b)所示, 即可在输出端的经典寄存器上得到不同精度的 λ^{-1} 。其中 λ^{-1} 的精度取决于迭代次数, 迭代次数越多, 精度越高。



(a) 一次迭代的倒数近似器电路



(b) n 次迭代的倒数近似器电路

图7 求倒数近似值的量子电路图

4 量子倒数器代价分析

4.1 时间复杂度

求取 n 位二进制数 λ 的倒数 $\frac{1}{\lambda}$, 需要用到量子全加器、量子乘法器、量子寄存器等模块, 该文设计的量子电路图分别如图2、图5和图7所示。由牛顿迭代法的电路图可知, 一共需要用到3次量子乘法器, 1次量子全加器, 而量子乘法器是基于量子全加器设计的, 不能单独讨论时间复杂度。以输入的 x_0 的位数和 λ 的位数相同来计算, 从最终的倒数器的电路来分析, 在求取 $2x_i$ 的时候, 需要用到2次加法操作和1次低位置零

操作。由全加器电路图可知, 单比特全加器需要2个 Toffoli 门和5个 CNOT 门, 因此 n 比特全加器一次加法需要 $7n$ 个通用量子门, 所以两次加法操作的时间复杂度为 $O(14n)$, 而低位置零操作不需耗费时间复杂度, 所以 $2x_i$ 的时间复杂度为 $O(n)$ 。同理可知, λx_0^2 的时间复杂度也为 $O(n)$, 所以 λx_0^2 的时间复杂度为 $O(n^2)$ 。由于 $-\lambda x_0^2$ 和 $2x_0$ 之间的加法时间复杂度小于乘法, 所以一步的倒数计算器最终的时间复杂度为 $O(n^2)$ 。用 n 个倒数器进行叠加相连, 时间复杂度为 $O(n^3)$ 。

4.2 空间复杂度

由图7所示, λ 与 x_0 都为 n 位二进制数的时候, 用

以存放 λ 与 x_0 的寄存器数量为 $2n$, 辅助寄存器 $|q_{\text{carry}_1}\rangle_1$ 至 $|q_{\text{carry}_n}\rangle_{2n-1}$ 的数量为 $2n^2 - n$, 寄存器 $|q_{\text{carry_sum}0}\rangle$ 和 $|q_{\text{carry_sum}1}\rangle$ 的数量一共为 $4n$, 寄存器 $|q_{\text{barrow}}\rangle$ 的数量为 $2n - 1$, 因此一共所需的寄存器为 $2n^2 + 7n - 1$ 。最终一次迭代的倒数计算器的空间复杂度为 $O(n^2)$ 。用 n 个倒数器进行叠加相连, 空间复杂度为 $O(n^3)$ 。

5 结束语

该文设计的基于牛顿迭代法的量子倒数计算器实现了求取 n 位二进制数的倒数, 并且改进了量子乘法器, 用低位置零操作代替移位操作。该设计大大降低了系统的空间复杂性。时间复杂性分析进一步表明, 所设计的量子倒数器的时间复杂度并未远大于单独的乘法器, 且结构简单, 易于优化, 迭代次数可根据具体精度需要任意选择。

参考文献:

- [1] 王书浩, 龙桂鲁. 大数据与量子计算[J]. 科学通报, 2015, 60(5): 499-508.
- [2] WITTEK P. Quantum machine learning: what quantum computing means to data mining[M]. New York: Academic Press, 2014.
- [3] YAO Zhang, QIANG Ni. Recent advances in quantum machine learning[J]. Quantum Engineering, 2020, 2(1): 1-20.
- [4] SCHULD M, SINAYSKIY I, PETRUCCIONE F. An introduction to quantum machine learning[J]. Contemporary Physics, 2015, 56(2): 172-185.
- [5] FEYNMAN R P. Quantum mechanical computer[J]. Foundations of Physics, 1986, 16(6): 507-531.
- [6] 黄一鸣, 雷航, 李晓瑜. 量子机器学习算法综述[J]. 计算机学报, 2018, 41(1): 146-148.
- [7] CAO Yudong, DASKIN A, FRANKEL S, et al. Quantum circuit design for solving linear systems of equations[J]. Molecular Physics, 2012, 110(15-16): 1675-1680.
- [8] CHENG Kai-Wen, TSENG Chien-Cheng. Quantum full adder and subtractor[J]. Electronics Letters, 2002, 38(22): 1343-1344.
- [9] 常丽, 朱宇祥, 蒋辉. 量子全加器设计[J]. 电子学报, 2019, 47(9): 1863-1867.
- [10] 袁素真, 王艳, 王玉婵, 等. 量子乘法器的设计及其实现方法[J]. 重庆邮电大学学报: 自然科学版, 2019, 31(3): 395-399.
- [11] SUZUKI Y, UNO S, RAYMOND R, et al. Amplitude estimation without phase estimation[J]. Quantum Information Processing, 2020, 19(2): 1-17.
- [12] KRYLOV A I, DOYLE J, NI KangKuen. Quantum computing and quantum information storage[J]. Physical Chemistry Chemical Physics, 2021, 23(11): 6341-6343.
- [13] LOPES J H, SOARES W C, BERNARDO B L, et al. Linear optical CNOT gate with orbital angular momentum and polarization[J]. Quantum Information Processing, 2019, 18(8): 1-10.
- [14] TOFFOLI T. Reversible computing[C]//7th international colloquium on automata, languages and programming. Noordwijkerhout, Netherlands: ICALP, 1980: 632-644.
- [15] ÁLVAREZ-SÁNCHEZ J J, ÁLVAREZ-BRAVO J V, NIETO L M. A quantum architecture for multiplying signed integers[J]. Journal of Physics: Conference Series, 2008, 128(1): 012013.
- [16] KOTIYAL S, THAPLIYAL H, RANGANATHAN N. Circuit for reversible quantum multiplier based on binary tree optimizing ancilla and garbage bits[C]//27th international conference on VLSI design and 2014 13th international conference on embedded systems. Mumbai, India: [s. n.], 2014: 545-550.
- [17] BABU H M H. Cost-efficient design of a quantum multiplier-accumulator unit[J]. Quantum Information Processing, 2017, 16(1): 1-38.